**Capstone Project 2: Milestone Report 1**

**Problem:**

As people travel from one city to another, they want to be able to experience aspects, cultures and peculiarities of that city. One of such distinctive features is the gastronomy, restaurants or eateries of that city. Being new to the city the average traveler will not know where to eat. As such his or her go-to will be Google or Yelp. Yelp would provide a list of restaurants closest to where the user of the app is, based on his GPS coordinates and not necessarily a list of restaurants that are popular in the city and that others just like the user would recommend. The objective here is to build a recommender system that takes eateries and restaurants within a city and recommends them based on how popular they are and also how likely they would be recommended to a user by other users of similar taste.

**Client:**

Business travelers, vacationers and yelp users who are foodies and like to travel. Restaurants will benefit from this as they get to be recommended not based on their distance from a user but rather their popularity and the likelihood that some other user similar in taste to the visiting user would recommend them.

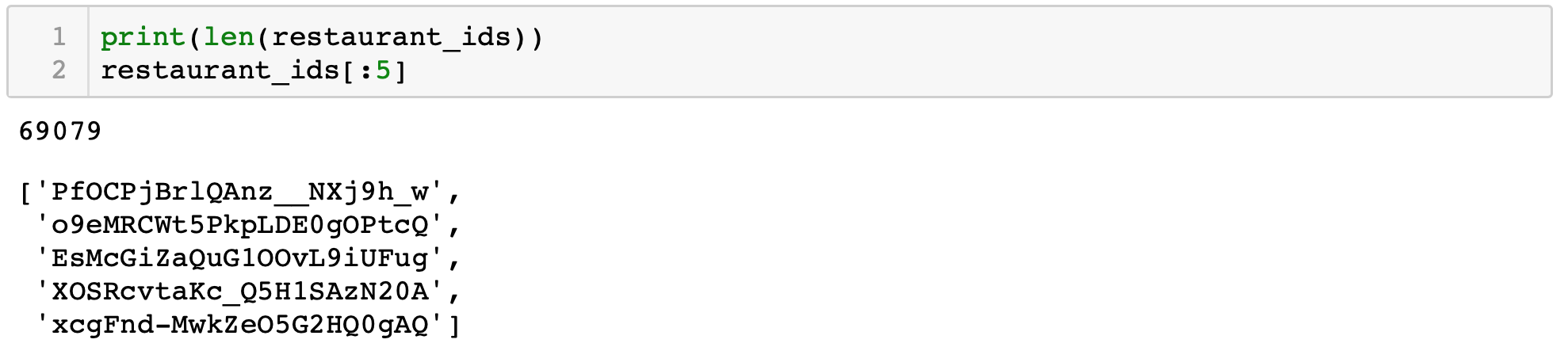
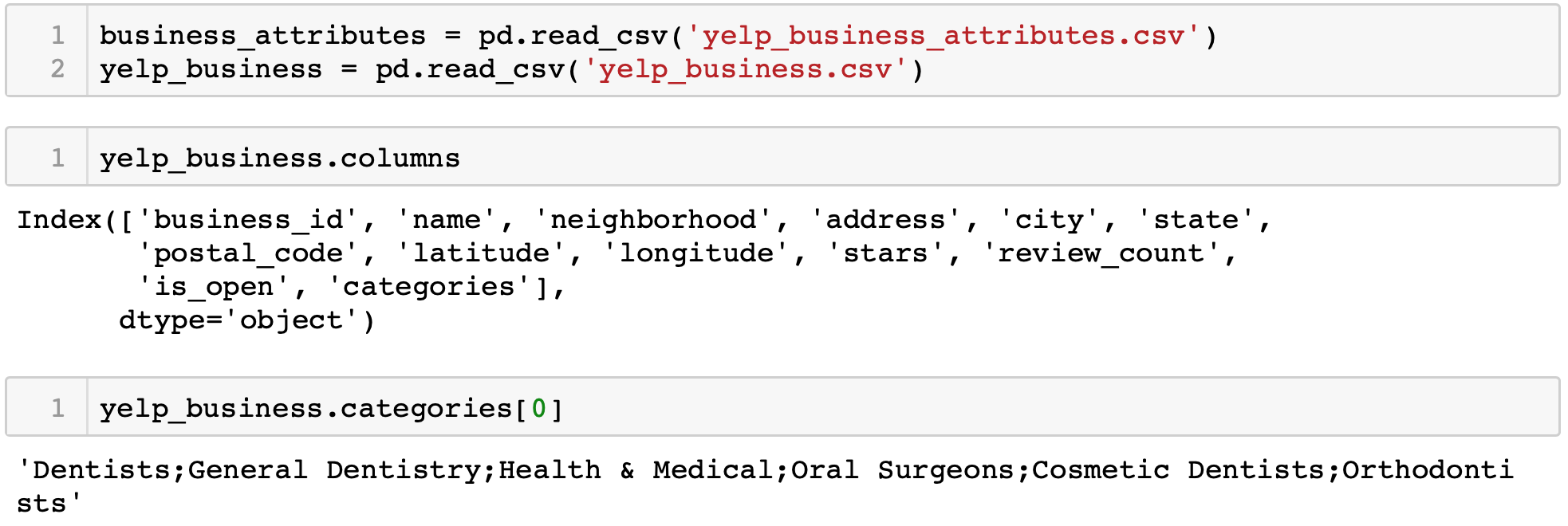
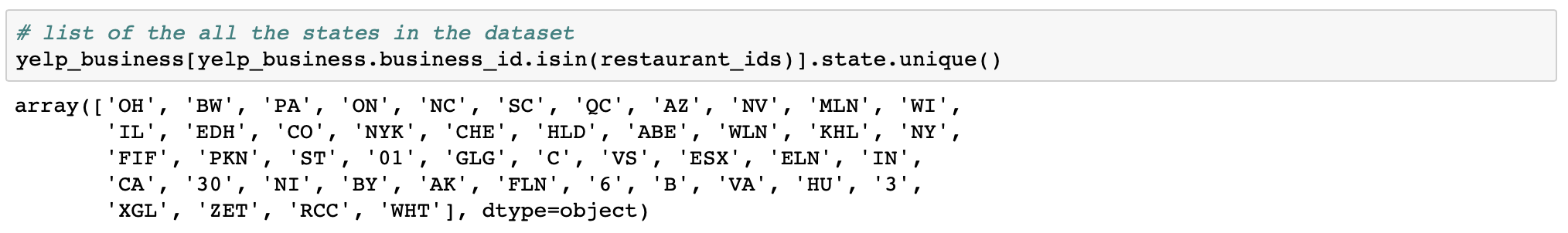
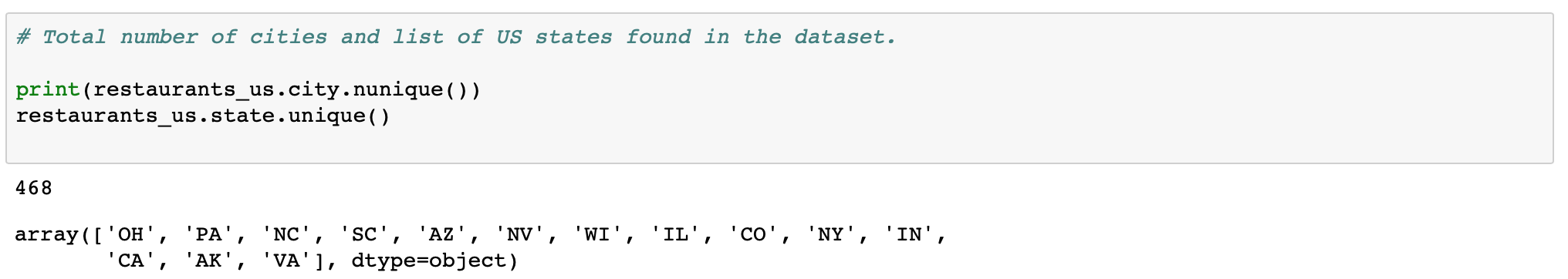
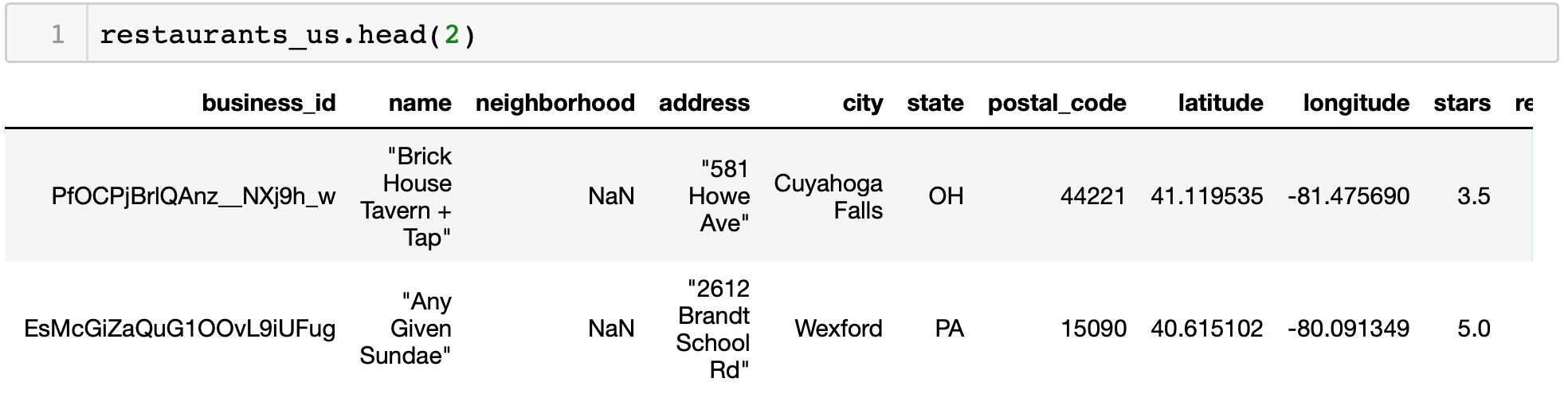
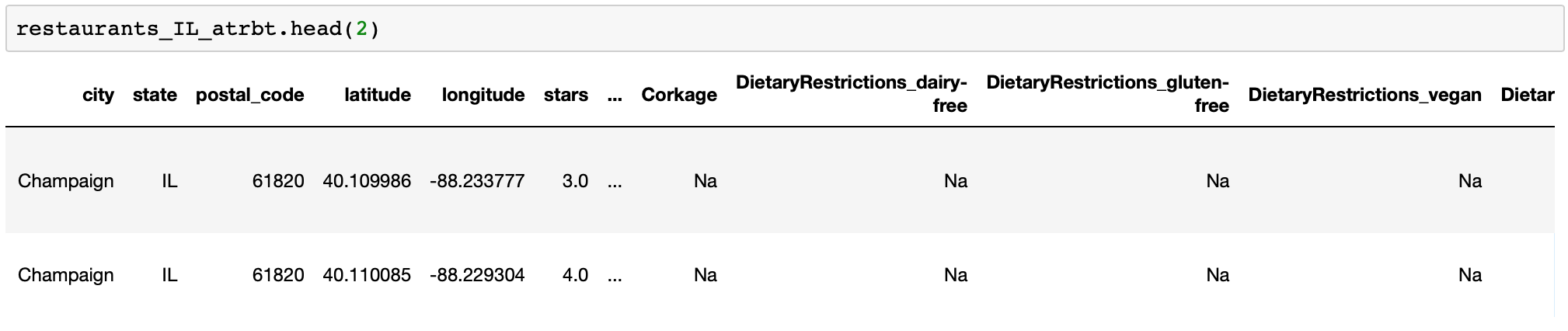
**Data Set:**

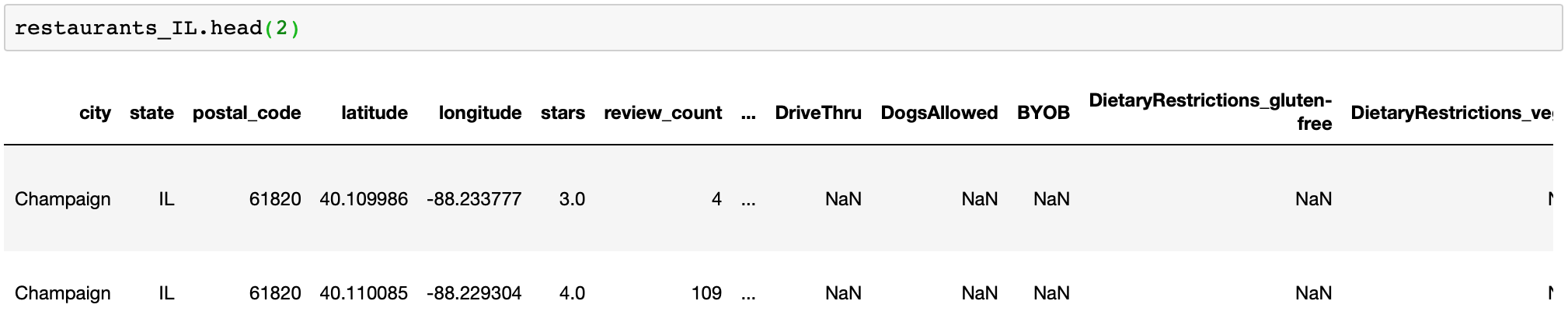
Yelp data set which has 1.637M users with 6.685M reviews for 192,609 businesses of all types in 10 metropolitan areas.

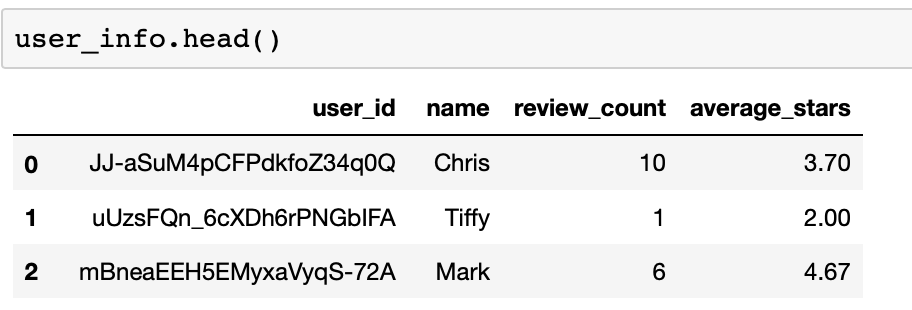
**Approach:**

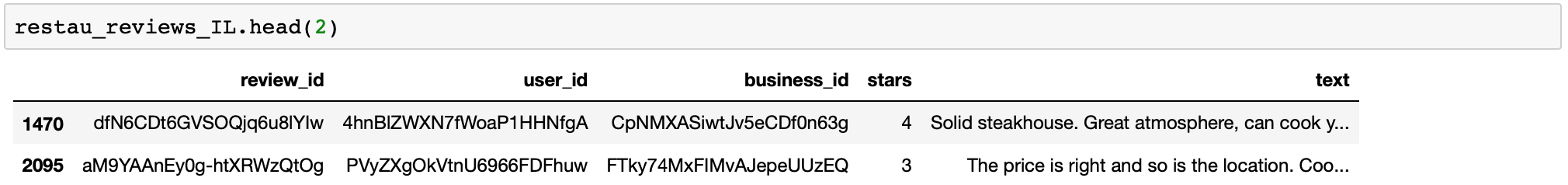
The dataset is made up of 4 different files yelp\_business.csv, yelp\_business\_attributes.csv, yelp\_user.csv and yelp\_review.csv. Used pandas to load the data from these files with its chunk feature coming handy for the very large files. Used the nltk library to determine from a string in the categories column of the yelp data set, which businesses were restaurants or eateries. And combined all the relevant data into a single table. Wrangled the data to eliminate missing values and also did some EDA on the ratings distribution of the users. The recommender is collaborative and the data set was narrowed down to restaurants in cities of Illinois found in the data set. Please note the reference to restaurants also refers to eateries.

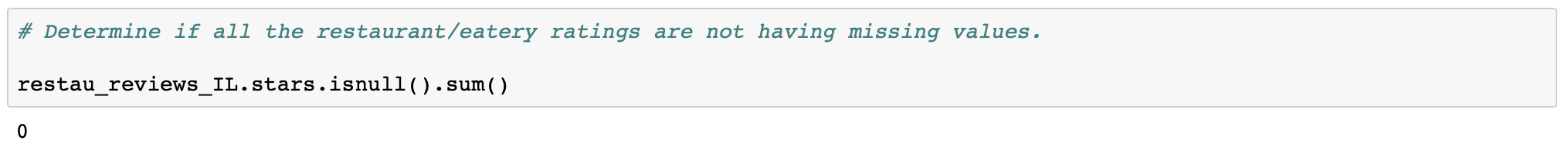
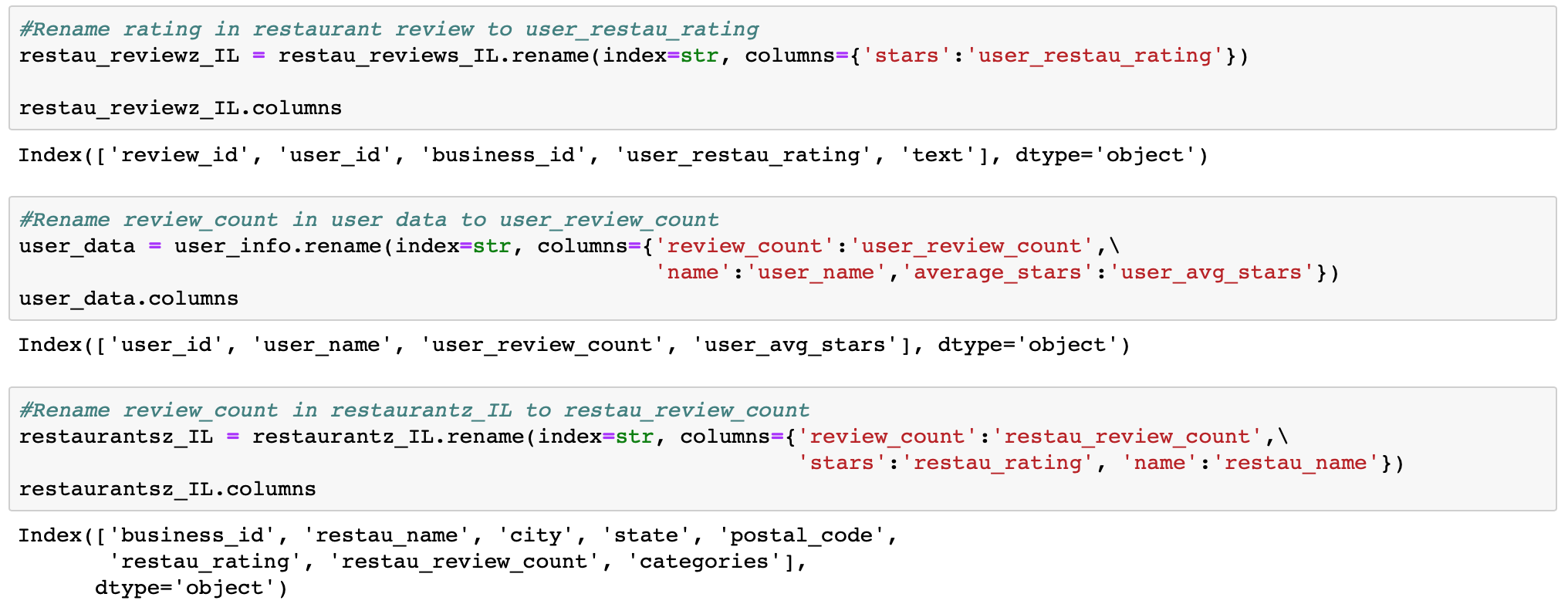
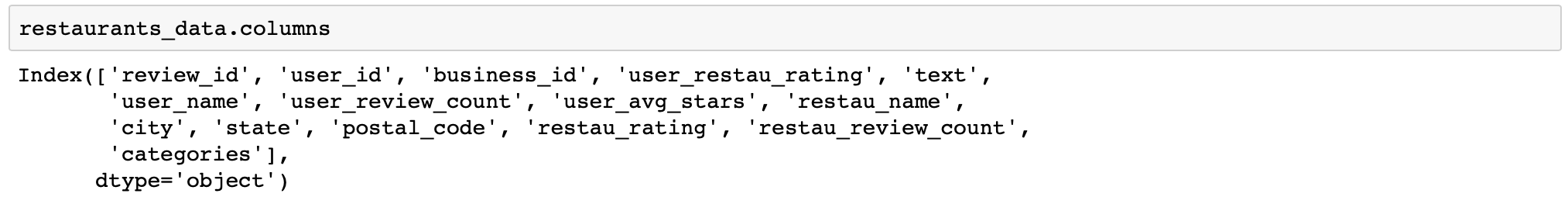
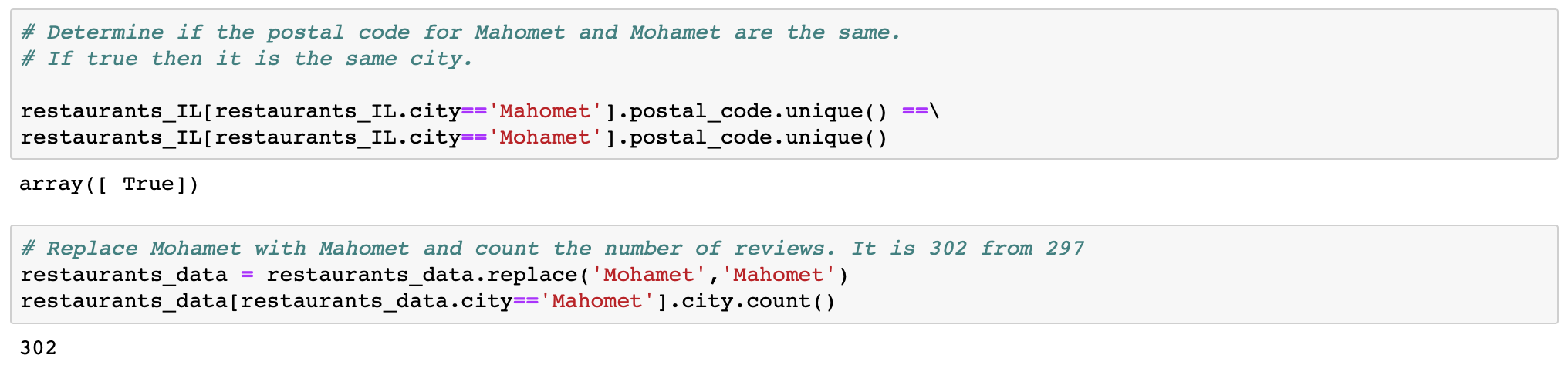
**Data Wrangling**

* Loading files in Pandas**:** 
  + yelp\_business.csv and yelp\_business\_attributes.csv files were loaded and the nltk library used on the “categories” column of the yelp\_business file to determine which businesses were restaurants or eateries and combined all the restaurant business ids in a list.
  + Examining the states found in the dataset
  + Loaded a txt file with US states name codes and combined with the list of restaurants ids to pull data from the yelp\_business data set of only restaurants in the US.
  + Pull from the dataset only restaurants found in Illinois and merge with the attributes file to add attributes to restaurants in Illinois. Change all Na entries to NaN entries.

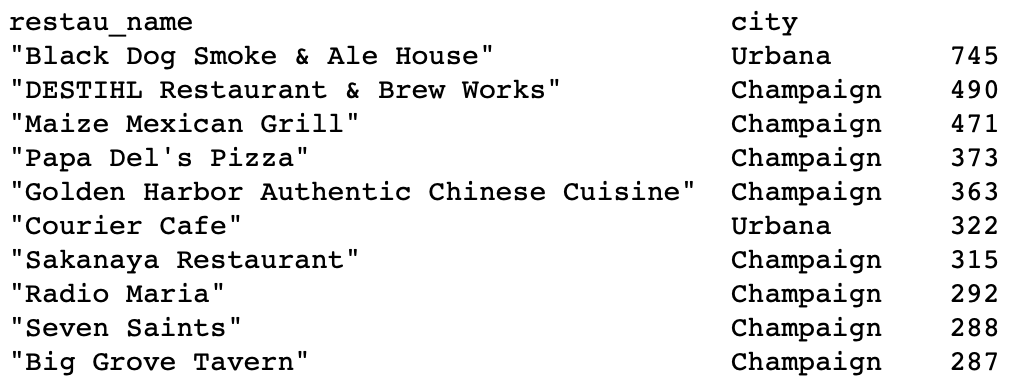


* + Load relevant user information from the 1.6GB yelp\_user.csv file in chunks of 10000 rows and also load from the 3.79GB yelp\_reviews.csv file reviews of users for restaurants in IL only.



* + Check for any missing ratings.
  + Before merging the files, rename columns having the same name between the files
  + Merge the three files into a single data frame restaurants\_data with column names thus:
  + Check City column and correct some misspelled city entries from Mohamet to Mahomet. Fill up missing postal codes and save data frame to restaurants\_all\_info\_IL.csv

**Data Story Telling:**

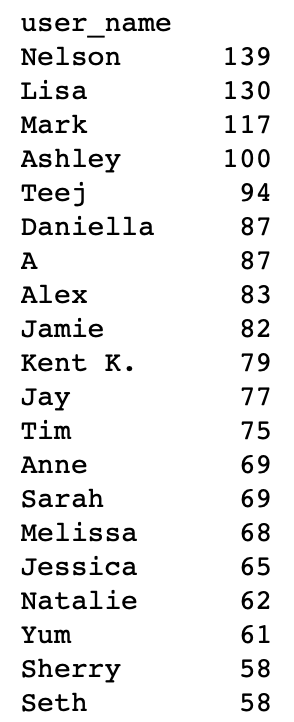
* + The most popular of the ratings given by the users for the different restaurants is 5 while the least popular is 2. This implies most users have
  + The highest number of the reviews were given for restaurants in the city of Champaign followed by the city of Urbana. This is primarily because most of the restaurants in the data set are found in Champaign and Urbana with 408 and 148 restaurants respectively.
  + The top 10 restaurants by number of reviews are found in Champaign and Urbana as shown below:

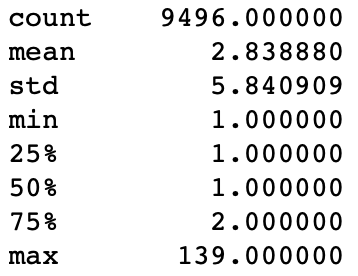
**Exploratory Data Analysis :**

The dataset used is made up of ratings from eateries in 19 cities in IL filtered from the yelp dataset. The former being a combination of user information, business attributes, business reviews and user reviews from 4 different files. The objective is to recommend to users, restaurants in the city that other users similar to them like and would recommend.

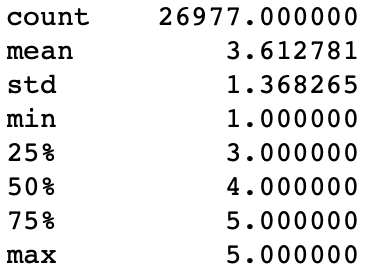
The main variables considered in the recommendation are the ratings given by the users.

The number of reviews by each user varies by a huge margin with about 95% of the users having about 10 or less reviews while 5% has between 10 and 140 reviews. This gives a right skewed distribution of number of reviews per user with a mean of approximately 3.0 per user and median of 1.0 as shown below:





This is an indication that most of the yelp users are not very active reviewers. You have 5% of the reviewers who are very active doing between 10 to 139 reviews per user and 95% who are not, providing 10 or less reviews per user.

A distribution of the ratings is left skewed with a majority of the ratings being 3 to 5 and a mean of 3.61 and median of 4.0.

**In-depth Analysis :**

The collaborative recommendation system is built on what other users have given as ratings to a particular item or in this case restaurant. Their ratings are used as inputs to recommend to other users who have not visited that restaurant before.

Given a set of restaurants with business ids, biz\_id\_1, biz\_id\_2, …, biz\_id\_6, and users with user ids, usr\_id\_1, usr\_id\_2, …, usr\_id\_10, the recommendation problem reduces to replacing the question mark entries in the ratings matrix below:

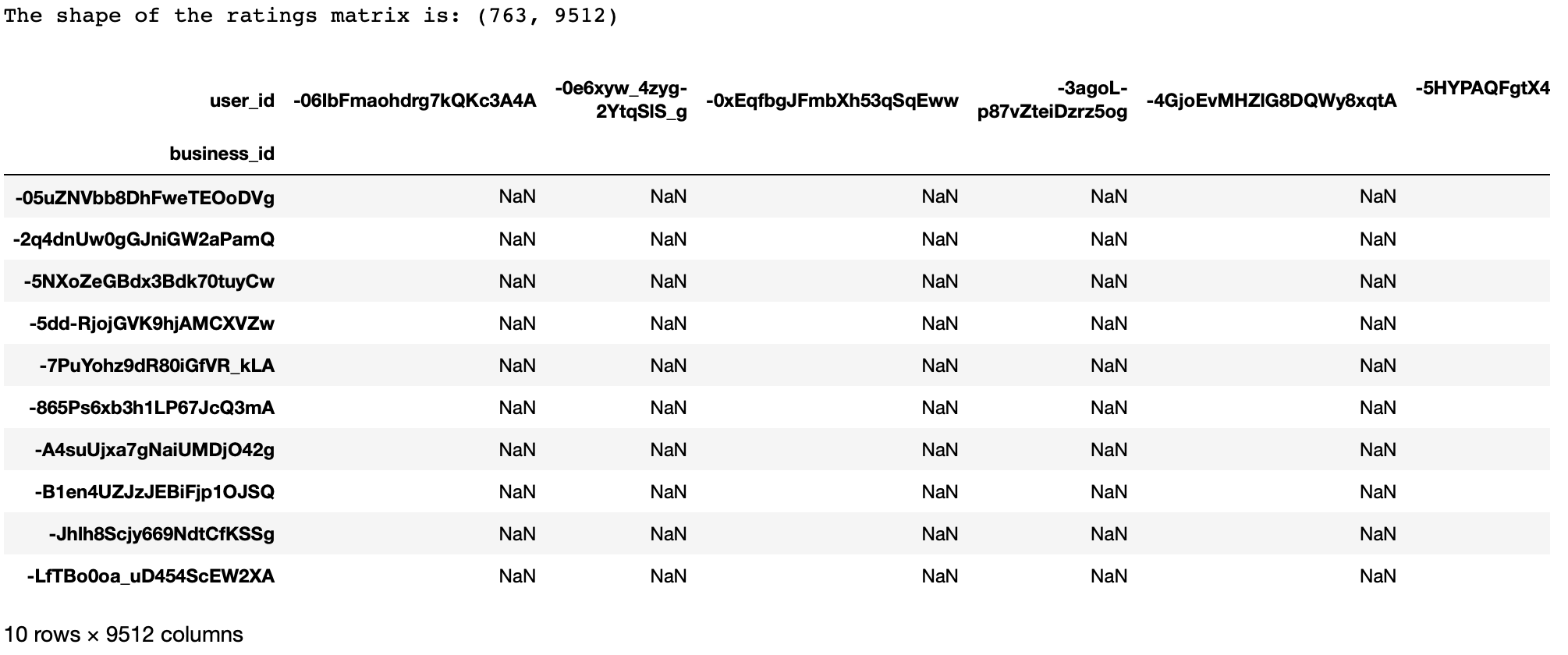
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| user\_id\business\_id | biz\_id\_1 | biz\_id\_2 | biz\_id\_3 | biz\_id\_4 | biz\_id\_5 | biz\_id\_6 |
| usr\_id\_1 | ? | ? | 3 | ? | ? | 5 |
| usr\_id\_2 | 2 | ? | 4 | 4 | 5 | 3 |
| usr\_id\_3 | 5 | 4 | ? | ? | 4 | ? |
| usr\_id\_4 | 5 | 1 | 5 | ? | ? | ? |
| usr\_id\_5 | ? | ? | ? | ? | 2 | 4 |
| usr\_id\_6 | ? | 5 | 4 | ? | 5 | 3 |
| usr\_id\_7 | 2 | ? | 1 | 5 | ? | ? |
| usr\_id\_8 | 5 | 3 | 4 | 3 | 2 | 5 |
| usr\_id\_9 | ? | ? | ? | 4 | 2 | 1 |
| usr\_id\_10 | 4 | 5 | 4 | ? | ? | 3 |

The ratings are filled by considering the ratings given by another user or a combination of other users. The assumption here is that, for a ratings prediction to be close to that the user in question would give, the other user or users should be very similar in their ratings profile of restaurants that this user has visited.

For instance we want to determine the rating of biz\_id\_5 by usr\_id\_10. The most similar user is usr\_id\_6. The system would recommend a rating of 5 stars for biz\_id\_5 by usr\_id\_10. Similarly using usr\_id\_2, a 4 stars rating for biz\_id\_4 is recommended for usr\_id\_6.

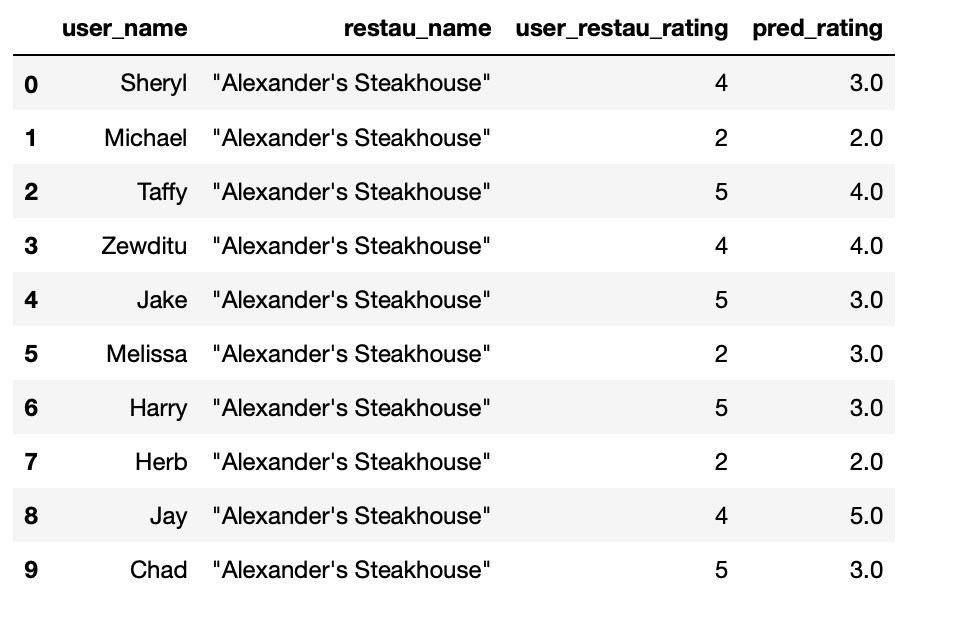
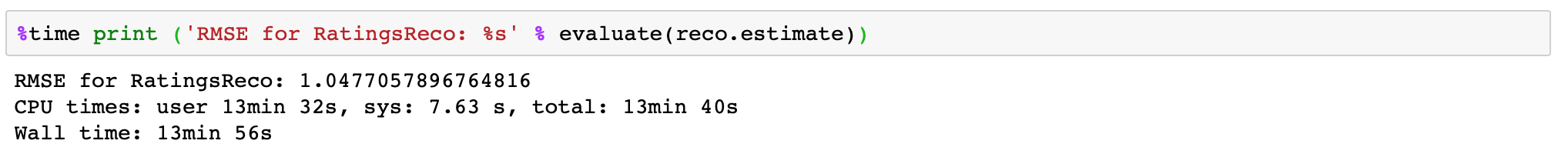
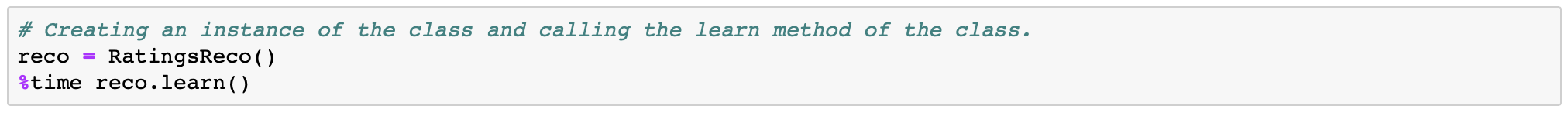
**Recommendation System**:

We develop a model using pandas with its data munging, wrangling and data modeling capabilities.The model makes use of the Pearson correlation between the users as the similarity metric. We take the average of the ratings of all users positively correlated to the specific user as the predicted rating. We define a class object “RatingsReco” that is composed of two methods “learn” and “estimate”. The “learn” method prepares the data structures, in particular the ratings matrix that is used by the “estimate” method to predict the rating. The ratings matrix for the restaurants and users is as below with the Nans representing the ? marks above.

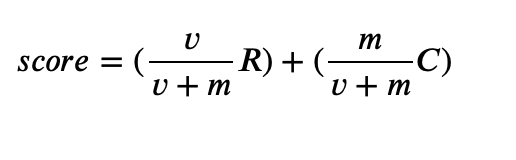
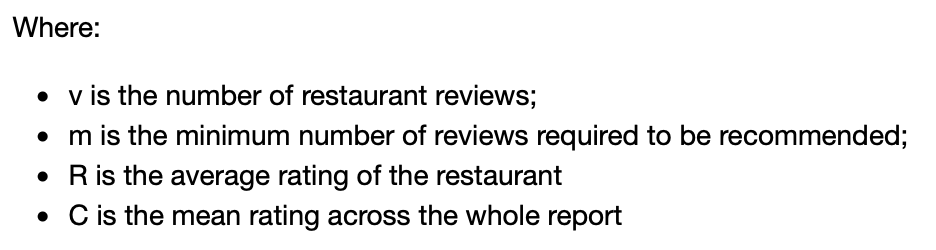


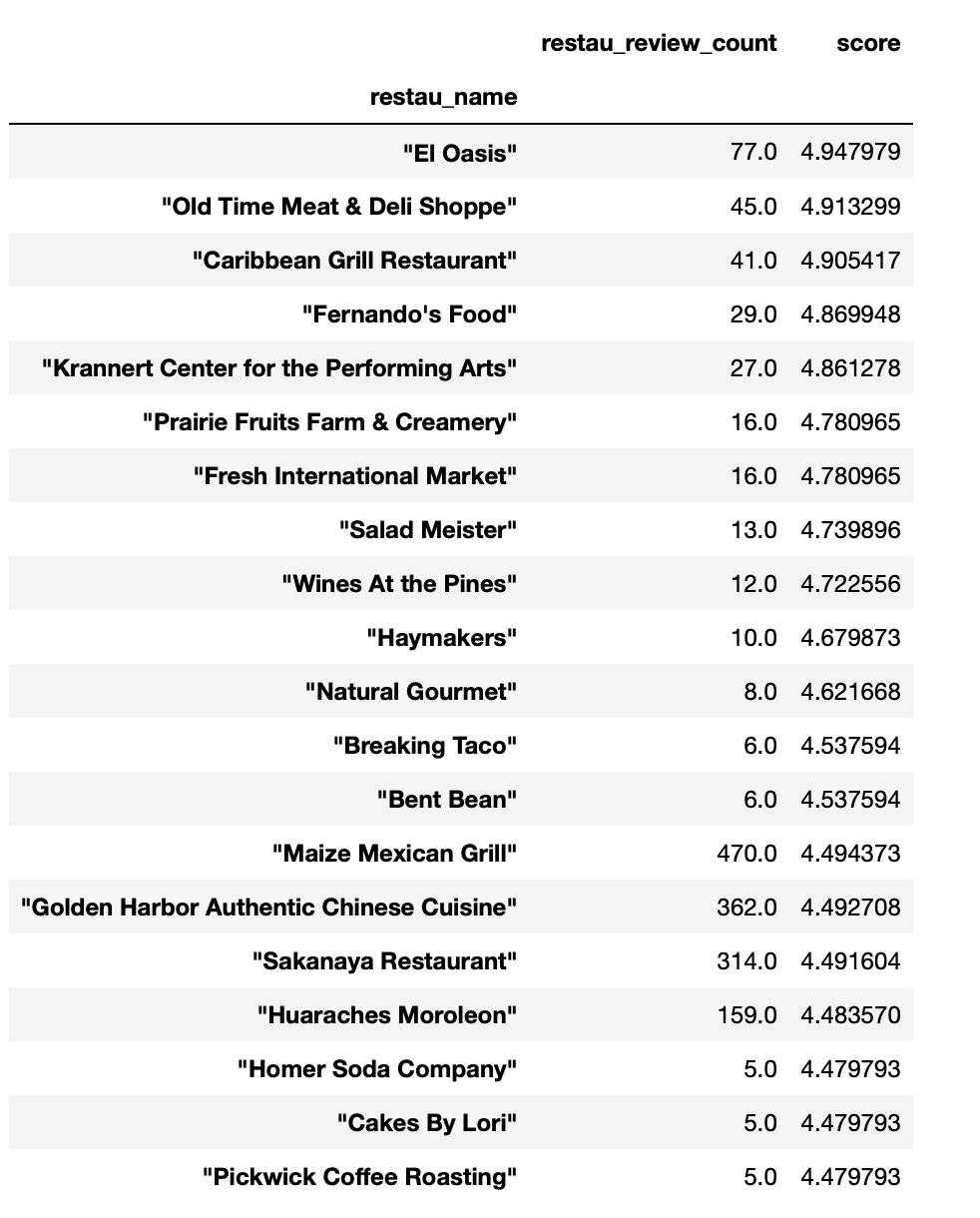
**Model Validation and Evaluation:**

To validate the model, we split the data using the train\_test\_split function from the sklearn.model\_selection library into a training and test data set in a ratio 4:1 respectively. The performance of the recommendation is measured using the “mean\_square\_error” function from the “sklearn.metrics" library. The training dataset is used to build the model after which it was applied to the test data set.

The **RMSE=1.05** which implies averagely the predicted ratings are approximately 1 star off the actual ratings which is quite a good prediction as shown below**.**

**Popularity Metric:**

In order not to recommend a restaurant rated 5 stars by one person over a restaurant rated 4.5 stars by 200 people, a weighted rating (score) is introduced. The “score” talking into consideration the overall rating of the restaurant (restau\_rating) and also the popularity of the restaurant (restau\_review\_count) i.e how many users have visited this restaurant. The score is calculated using the IMDB formula:



We have the following:

**City Restaurant Recommendation:**

To make sure we are recommending the restaurants that are popular with a 5 star predicted rating, we take an average of the score and the predicted ratings to recommend restaurants to users from one city to another as shown below. Amanda is recommended 6 additional eateries in the city of Champaign when compared to J. and Mindy

**Conclusion:**

This recommendation system provides a user with a list of restaurants in a city that other users similar in taste to this specific user would recommend to him.

A specific case of 19 Illinois cities extracted from the yelp data set was used. This could be scaled to different cities nationally.

Such a system is useful in recommending restaurants to a yelp user when he gets into a new city not based on distance but on its popularity and rating by other users similar to this user.

**Recommendation:**

* + This analysis was purely collaborative using explicit data (in this case ratings) and as such would not offer any recommendations in a restaurant cold start problem. In that case, a content based recommendation system which looks at the similarity in the attributes of one restaurant compared to another reviewed by the user would be a solution to a more personalized rating. Using the business attributes as independent variables and the rating as the dependent variable a Regression model could be used.